Использование сверточных нейронных сетей для распознавания трехмерных объектов в реальном времени

С.А. Разлацкий[a], П.Ю. Якимов[a]

[a] *Самарский национальный исследовательский университет имени академика С.П. Королева, 443096, Московское шоссе, 34, Самара, Россия*

Аннотация

Задача распознавания объектов в реальном времени является ключевой для автономного функционирования интеллектуальных систем компьютерного зрения. Различные 3D сенсоры, такие как LiDAR или стереокамеры, становятся все более распространены в современных робототехнических системах и являются надежным источником трехмерных данных. Однако, многие современные системы используют получаемые данные неэффективно и сталкиваются с проблемой недостаточной производительности. В настоящей статье исследованы современные подходы к распознаванию трехмерных объектов в реальном времени, в том числе с использованием трехмерных сверточных нейронных сетей. Для оценки качества распознавания и быстродействия рассмотренных методов были использованы различные типы трехмерных данных.

Ключевые слова: сверточные нейронные сети; распознавание трехмерных объектов; обработка в реальном времени; глубинное обучение; трехмерные изображения

1. Введение

Задача надежного детектирования объектов в трехмерной сцене стала актуальной с появлением и развитием методов получения трехмерных цифровых изображений. Между тем, сенсоры, такие как LiDAR и RGBD камеры, развивались и становились все более распространённым решением для многих автономных робототехнических средств. Настоящая работа является продолжением исследований методов распознавания трехмерных объектов в облаке точек – специализированном формате представления трехмерных данных.

Методы, рассмотренные авторами в [1] и [2], эффективно распознают объекты, и их быстродействия достаточно для функционирования в реальном времени, но главным недостатком описанных методов является требование идентичности искомого объекта заранее заданному эталону. То есть, для распознавания какого-либо зарегистрированного объекта необходимо, чтобы в системе уже имелась трехмерная модель, ассоциированная с определенным классом. Такой подход хорошо подходит для распознавания объектов на модельных данных, но в реальности регистрируемый объект часто отличается от его представления в эталонных моделях.

В последнее время все более широкое распространение получили методы на основе технологии глубинного обучения. В частности, задача детектирования и распознавания трехмерных объектов успешно решается при помощи сверточной нейронной сети (ConvolutionalNeuralNetwork или CNN), адаптированной для работы с трехмерными объектами [3]. Настоящее исследование посвящено изучению работы библиотеки VoxNet [4], которая содержит эффективную реализацию трехмерной сверточной нейронной сети (3DCNN) для распознавания трехмерных объектов в реальном времени.

2. Обзор существующих методов

2.1. Классические методы распознавания трехмерных объектов

Одним из примеров классического подхода к обнаружению трехмерного объекта является трехмерный метод Хафа [5]. Данный метод стал популярен в применении для двумерных изображений, где в качестве областей интереса использует в основном контуры объектов. При работе в трехмерном формате метод выделяет в трехмерном изображении особые точки для сокращения вычислительных издержек. Такие точки выделяются при помощи специального алгоритма кластеризации. Процедура голосования происходит в аккумуляторном пространстве с учетом только выделенных особых точек. В результате получаются локальные максимумы в тех областях, где потенциально может находится искомый объект. Еще одна сложность с добавлением третьего измерения, помимо возросшей вычислительной нагрузки, – это вероятность различной ориентации сцены и искомого объекта. Эта проблема решается при помощи введения специальных векторов, обеспечивающих инвариантность к вращению и повороту [1]. Немного более проработанным является метод геометрической связности [2]. Основным отличием от трехмерного метода Хафа является иной алгоритм поиска особых точек, которые объединяются в так называемые особые области и переводятся в формат, описываемый специальным индексом форм. Полученные области в виде значений индексов записываются в двумерные гистограммы, где и происходит процедура голосования по всем локальным окрестностям, содержащихся в тестовом объекте.

Существует большое количество работ по распознаванию объектов в трехмерном облаке точек, получаемых при помощи LiDAR и стереокамеры, использующих комбинацию различных индивидуальных признаков и дескрипторов с классификацией методами машинного обучения [6], [7], [8]. Также широко распространены методы семантической сегментации, где вместо отдельные классификаторы используются структурированные классификаторы. В отличие от

указанных выше подходов, исследуемая в настоящей работе архитектура учится извлекать особенности и классифицировать объекты из «сырых» 3D данных. Объемное представление также лучше, чем облака точек, тем как оно отличает свободное пространство от неизвестного. Кроме того, методы с использованием облаков точек требуют для вычислений окрестности точек, что часто становится вычислительно неразрешимыми с большим количеством точек.

2.2. 2.5D CNN

Вдохновившись успешным применением сверточных нейронных сетей для решения задач распознавания на двумерных изображениях, некоторые авторы расширили их использование для стерео данных. Такие подходы обрабатывают канал с «глубиной» как дополнительный канал, наряду с обычными каналами R, G, B. Однако, при этом не в полной мере используется геометрическая информация в трехмерных данных, что затрудняет интеграцию между зрительными точками.

Для LiDAR данных предложены признаки [9], локально полученные на данных с представлением 2.5D, а некоторые работы исследуют данный подход в сочетании с разновидностью так называемого обучения без учителя [10]. В работе [11] предложена кодировка, которая эффективно использует информацию о глубине, но подход все равно двумерно-ориентированный. Получается более точное представление об окружающей среде.

3. VoxNet. Архитектура 3D CNN

Исходные данные для алгоритма, реализованного в библиотеке VoxNet, представляют собой сегмент облака точек, который может быть получен различными методами сегментации или при помощи алгоритма «скользящего трехмерного окна». Сегмент, как правило, определяется пересечением облака точек с отграничивающим параллелепипедом и может включать в себя фоновые шумы. Задача заключается в определении принадлежности объекта для данного сегмента к определенному классу. Система решения данной задачи состоит из двух компонентов: «объемной сетки», которая представляет оценку пространственного наполнения, и 3DCNN, которая классифицирует объекты, непосредственно используя объемную сетку. Опишем компоненты более подробно.

На рисунке 1 представлена архитектура VoxNet.

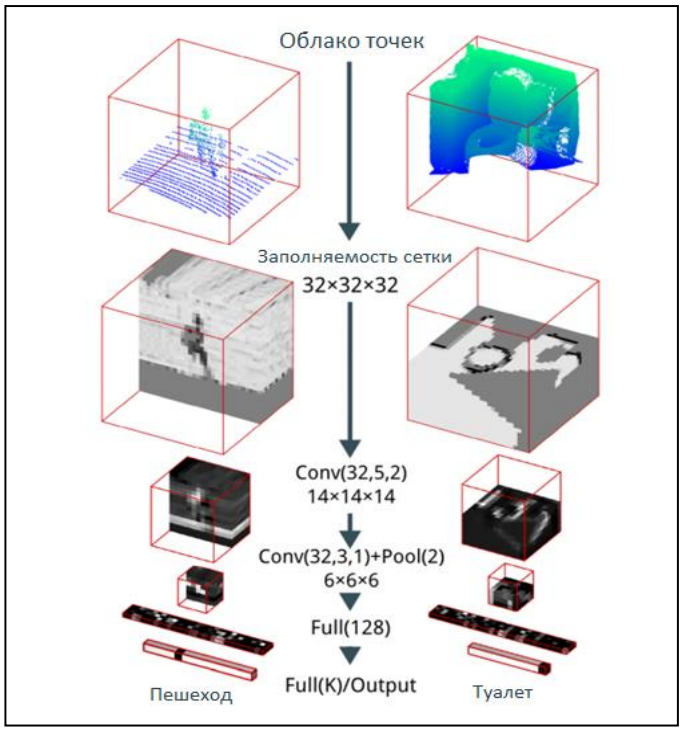

Рис. 1. Conv(f,d,s) указывает на фильтры F размера d и с шагом s, Pool(m) указывает на объединение площадью m, а Full(n) указывает на полностью связанный слой с n выходами. Слева представлен набор данных типа LiDAR, справа RGBD.

3.1. Сетка пространственного наполнения

Сетка пространственного наполнения отображает представление состояние среды как 3D решетки случайных величин (каждая соответствует одному вокселу) и формирует вероятностную оценку их размещения, как функцию от входных данных датчика и априорного знания. Есть две основных причины, по которым используются сетки пространственного наполнения. Во-первых, они позволяют эффективно оценить свободные, занятые и неизвестные пространства из измерений дальности, даже для измерений, поступающих с разных точек зрения и в различные моменты времени. Это представление лучше, чем те, которые рассматривают пространство как свободное или занятое.

Во-вторых, они могут быть сохранены и управляются с помощью простых и эффективных структур данных. В работе [3] используются плотные массивы, чтобы выполнить все CNN обработки, так как используются небольшие объемы данных (32^3 вокселя), а GPU эффективно работают с плотными данными. Для того чтобы сохранить весь объем данных, используются иерархические структуры данных и копирования конкретных сегментов в плотные массивы по мере необходимости. Теоретически это позволяет хранить потенциально неограниченный объем данных при использовании небольшой заполняемой сетки для обработки CNN.

В объемном представлении каждая точка (x,y,z) сопоставляется с дискретными координатами вокслей (I,J,K). Отображение равномерной дискретизации зависит от происхождения, ориентации и разрешения вокселей сетки в пространстве. Появления вокселезирванных объектов в значительной степени зависит от их параметров. Пусть $\{z^t\}_{t=1}^T$последовательность измерения дальности, которые либо попали ($z^t = 1$) или прошли через ($z^t = 0$) заданный воксель с координатами (I,J,K). Предполагая, что имеется идеальный датчик света, используется трассировка трехмерных лучей, чтобы выяснить количество ударяющихся и проходящих лучей для каждого вокселя. Учитывая эту информацию, рассматривается три различные модели наполнения пространственной сетки. В бинарном размещении каждый воксель предполагает бинарное состояние: занятости или незанятости. Вероятностная оценка занятости для каждого вокселя вычислена как логарифм отношения шансов (logodds) для численной стабильности. Обновляется каждый воксель проходящий через луч следующей формулой:

$$l_{i,j,k}^t = l_{i,j,k}^{t-1} + z^t l_{occ} + (1 - z^t) l_{free} \quad (1)$$

Где l_{occ}и l_{free} являются вероятностью клетки, занимаемых или не занимаемых, при условии замера удара или пропуска клетки, соответственно. Устанавливается их значения как $l_{occ} = 1{,}38$ и $l_{free} = -1{,}38$ и зажатие вероятностиlogв (-4,4), чтобы избежать численных проблем. Эмпирически было обнаружено, что в разумных пределах параметры не оказывают сильного влияния на конечный результат. Начальная вероятность размещения устанавливается до 0,5 или $l_{i,j,k}^0 = 0$. В этом случае сеть действует на значения $l_{i,j,k}$.

В методе плотности сетки предполагается, что каждый воксель имеет непрерывную плотность, которая соответствует вероятности вокселя, блокирующий датчик света. Используются формулы, где параметры $\alpha_{i,j,k}^t$и$\beta_{i,j,k}^t$, с равномерно априорными $\alpha_{i,j,k}^0$=$\beta_{i,j,k}^0$=1 для всех (i,j,k). Отсюда получаются следующие формулы:

$$\alpha_{i,j,k}^t = \alpha_{i,j,k}^{t-1} + z^t$$

$$\beta_{i,j,k}^t = \beta_{i,j,k}^{t-1} + (1 - z^t)$$

Апостериорные значения для (i,j,k) получается:

$$u_{i,j,k}^t = \frac{\alpha_{i,j,k}^t}{\alpha_{i,j,k}^t + \beta_{i,j,k}^t} \quad (2)$$

В этом случае получается $u_{i,j,k}^t$в качестве входных данных в сеть.

Метод попадание сетки рассматривает только попадания, и игнорирует разницу между свободным и неизвестным пространством. Каждый воксель имеет начальное значение $h_{i,j,k}^0$=0и обновляется так:

$$h_{i,j,k}^t = min\left(h_{i,j,k}^{t-1} + z^t, 1\right) \quad (3)$$

Хотя эта модель отбрасывает потенциально полезную информацию, в проведенных экспериментах она проходит довольно успешно. Более того, он не требует трассировки лучей, что полезно в вычислительно ограниченных ситуациях.

3.2. 3D CNN

Есть три основных причины, делающих CNNs привлекательным для решения поставленной задачи. Во-первых, они могут использовать пространственную структуру рассматриваемой задачи. В частности, они могут изучить локальные пространственные фильтры, полезные для задачи классификации. В данном случае, ожидается, что фильтры на уровне входного сигнала будут кодировать пространственные структуры как плоскости и углы при различных ориентациях. Во-вторых, укладывая несколько слоев сети можно построить иерархию из более сложных функций, представляющих большие области пространства, в конечном итоге это приводит к полной классификации для поступающей сетки пространственного наполнения. И наконец, такие операции могут быть выполнены эффективно на графическом аппаратном средстве. В данном исследовании рассматривается CNN, состоящая из нескольких слоев.

Входной слой принимает фиксированного размеры сетки I*J*K вокселей. В выбранной версии используются вариант, где каждый параметр равен 32. В зависимости от выбранного метода пространственного заполнения, значение каждой ячейки находится в диапазоне (-1,1), основываясь на уравнениях или (1), или (2), или (3). Никаких дальнейших предварительных обработок не требуется. Хотя эта работа рассматривает только скалярный вклад, реализация может

принимать дополнительные значения в ячейку, такие как LiDAR значения интенсивности или RGB информация от камер.

Сверточные слои C(f,d,s) при помощи фильтра f, размерности исследуемого блока вокселей d и шагом s, создают карту признаков f.Создание карты признаков происходит путем перемножения входных данных с фильтром. При переходе к сверточным слоям следующего уровня, начинается поиск более сложных признаков. Для повышения эффективности они опираются на результаты карт признаков предыдущего уровня, получая четырехмерный объем данных d*d*d*f', где d является пространственным размером, а f' – карты признаков предыдущего уровня. Свертка также может быть применена при большом пространственном шаге s, что может повысить скорость обучения сети, но также может понизить точность распознавания. Полученный результат пропускается через выпрямленный нелинейный блок (ReLU) с параметром 0,1.

Группировка слоев P(m) подразумевает под собой нелинейное уплотнение карты признаков с шагомm по каждому трехмерному параметру (I,J,K). То есть вместо блока вокселей размера m*m*m мы получаем максимальное значение вокселя, представленного в матрице.

Полносвязанные слои FC(n) имеют n выходных нейронов. Выход каждого нейрона изучает линейная комбинация всех выходов предыдущего слоя, пропуская их через нелинейность. Для этого используется ReLUs, которая преобразовывает результаты выходного слоя для получения вероятности нахождения объекта, где число выходов соответствует числу классов.

Стоит понимать, что при первых попытках обнаружения искомых объектов сеть будет мало эффективна, основная причина – это принятия веса карты каждой карты признаков случайным образом. CNN начинает превосходить все ранее исследуемые методы лишь после обучения. Это обучение основывается на стохастическом градиентном спуске. Именно данный метод начинает регулировать вес карт признаков, приводя с каждым тренировочным набором данных данный параметр к эталонному значению.

4. Эксперименты

Чтобы оценить эффективность VoxNet, используются данные трех разных типов: LiDAR облака точек, RGBD облака точек и CAD модели. Рисунок 2 показывает примеры каждого типа.

Рис. 2. Сверху вниз. Облако точек из Сиднейских объектов, облако точек из NYUv2, и две вокселизированные модели ModelNet40.

LiDAR данные – Городские объекты Сиднея. Первый эксперимент проводился на городских объектах Сиднея, которые содержат меченые LiDAR отсканированные городские объекты в 26 категориях. Был выбран этот набор данных для оценки, поскольку он обеспечивает меченые экземпляры объекта и LiDAR точку наблюдения, которая используется для вычисления вместимости. Когда вокселизируется облако точек, используются все точки в ограниченной рамке вокруг объекта, в том числе на фоне помех. Для того, чтобы сравнивать результаты с другими работами, исследуем протокол, используемый авторами других наборов данных. Сообщается средний балл F_1, взвешенный по поддержке класса, для подмножества 14 классов в течении четырех стандартных расколов обучения/тестирования. Для этого набора данных проводится увеличение и голосование с 18 оборотами в инстанции.

CAD данные – ModelNet: ModelNet данные были введены автором Wu [12] для оценки 3D классификаторов формы. ModelNet имеет 128 3D моделей которые подразделяются на 40 категорий. Авторы предоставляют 3D-модели, а также вокселизированные версии, которые были дополнены 12 вращениями. Используется представленная вокселезация и

тест расщепления для оценки. Эти вокселизированные объекты масштабированы таким образом, чтобы соответствовать сетке 30*30*30; поэтому не ожидается польза от мультимасштабного подхода и используется VoxNet с одним разрешением. Для сравнения производительности сообщается средняя точность в каждом классе.

RGBD данные – NYUv2: Wu также оценивает подход на RGBD облаке точек, полученных из NYUv2 набора данных [13]. Используется раздельное испытание, предоставленное авторами, которые используют 538 изображений с вызовом RMRC для обучения, а остальные для тестирования. После отбора данных, которые содержат те же классы, что и ModelNet10, получается 1386 пространств для тестирования и 1422 объектов для обучающих. Для этого набора данных вычисляется собственные сетки пространственного наполнения. Однако, чтобы получить результаты сравнимые с Wu, не используется фиксированный размер воксела; вместо этого вырезается и масштабируется объект, ограниченный размером 24*24*24, с четырьмя полями вокселей; точно так же, используется 12 ротаций вместо 18. Как и в Сиднейском наборе данных, все точки держатся в ограничительной рамке вокруг объекта.

В ходе экспериментов были использованы данные, дополненные вращением. Было рассмотрено 4 различных случая для такого дополнения (аугментации) вращением: в зависимости от того, применяется ли оно или нет при обучении (как дополнение) и тестировании (как голосование) для сиднейских объектов и ModelNet40. Для случаев, в которых голосование не выполняется при тестировании, случайная ориентация применяется на тестовых примерах, и берется среднее время за 4 выполнения. Для случаев, в которых аугментация не выполняется, существует два варианта. В ModelNet40 для обучения выбирается объект в начальной позе. Для Сиднейских объектов эта информация не доступна, и используется не модифицируемая ориентация для данных. Таблица 1 показывает результаты экспериментов. Они показывают, что тренировочное время аугментации наиболее важно.

Таблица 1. Эффект от аугментации вращением и голосования

Аугментация при обучении	Голосование при тестировании	Сиднейские объекты	ModelNet40
Да	Да	0,72	0,83
Да	Нет	0,71	0,82
Нет	Да	0,69	0,69
Нет	Нет	0,69	0,61

Помимо этого, было произведено сравнение VoxNet с другим методом ShapeNet, предложенным Wu [12] для задачи классификации данных ModelNet40 и ModelNet10. В ShapeNet также используется объемная сверточная архитектура с применением аугментацией вращения для обучения. Однако количество параметров в этой архитектуре более 12 миллионов, как в VoxNet их менее миллиона. В таблице 2 показаны полученные результаты.

Таблица 2.Сравнение с ShapeNet

Набор данных	ShapeNet	VoxNet
ModelNet10	0,84	0,92
ModelNet40	0,77	0,83

Для обучения 3D сверточной нейронной сети и непосредственного выполнения был использован графический процессор NvidiaTeslaK40. Наиболее ресурсозатратная конфигурация VoxNet – мультимасштабность и дополнение поворотами и голосованием – выполняется за 6 мс в случае индивидуальной классификации, и по 1 мс на объект, если обрабатывать сразу 32 объекта. Такая разница объяснима накладными расходами на пересылку данных между GPU и CPU.

5. Заключение

В настоящей работе был представлен анализ библиотеки VoxNet, реализующей архитектуру трехмерных сверточных нейронных сетей для эффективного и точного обнаружения и распознавания объектов различных типов в облаках точек, а также изучено влияние использования различных конфигураций на производительность системы. Кроме того, было произведено сравнение с аналогичными системы на одинаковых наборах данных, которое показало превосходство исследуемой системы VoxNet. Полученные результаты показали, что данная реализация 3DCNN успешно справилась с поставленными задачи и действует эффективнее других подобных систем.

В дальнейшем планируется применить полученные результаты для исследования распознавания объектов в трехмерных сценах, получаемых при помощи стереокамеры, в реальном времени.

Благодарности

Исследования проводились при поддержке фонда РФФИ (проект 16-37-60106).

Литература

[1] Разлацкий, С.А. Применение метода Хафа для детектирования объектов в трехмерной сцене / С.А. Разлацкий, П.Ю. Якимов // Информационные технологии и нанотехнологии (ИТНТ-2015). -2015.

[2] Разлацкий, С.А. Применение метода геометрической связанности для детектирования объектов в трехмерном облаке точек / С.А. Разлацкий, П.Ю. Якимов // Информационные технологии и нанотехнологии (ИТНТ-2016). -2016.
[3] Nielsen, M. Neural networks and deep learning / M.Nielsen. – 2014.
[4] Maturana, D. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition / D. Maturana, S. Scherer // IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2015.
[5] Goshin, Ye.V. Segmentation of stereo images with the use of the 3D Hough transform / Goshin Ye.V., Loshkareva G.E. // CEUR Workshop Proceedings. – Vol. 1638 – 2016. – P.340-347.
[6] Tombari, F. Object recognition in 3D scenes with occlusions and clutter by Hough voting / L. Di Stefano, F. Tombari// Fourth Pacific-Rim Symposium on Image and Video Technology. – 2010. – P. 2-4
[7] Frome, A. Recognizing objects in range data using regional point descriptors / A. Frome, D. Huber, and R. Kolluri // ECCV. - 2004. - Vol. 1. - P. 1–14.
[8] Chen, H. 3D free-form object recognition in range images using local surface patches / H. Chen, B. Bhanu // Pattern Recognition Letters. – vol. 28(10). – 2007. – P. 1252-1262.
[9] Quadros, A. An occlusion-aware feature for range images / A. Quadros, J. Underwood, and B. Douillard // ICRA, May 14-18 2012.
[10] Quadros, A. Unsupervised feature learning for classification of outdoor 3d scans / M. De Deuge, A. Quadros, C. Hung, and B. Douillard // ACRA, 2013.
[11] Gupta, S. Learning rich features from RGB-D images for object detection and segmentation / S. Gupta, R. Girshick, P. Arbelaez, and J. Malik // ECCV, 2014.
[12] Wu, Z. 3d shapenets: A deep representation for volumetric shape modeling / Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao // Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition. – 2015.
[13] Silberman, N. Indoor segmentation and support inference from rgbd images / N.Silberman, D. Hoiem and R. Fergus // ECCV. – 2012.